

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

- ¿Qué es GitHub?

Es una comunidad donde poder compartir tus repositorios y similar a una red social podemos ver, seguir y interactuar con los repositorios de otros perfiles

- ¿Cómo crear un repositorio en GitHub?

En la parte superior a la derecha presionamos el "+" y le damos a "Create new repository", escribimos el nombre del repositorio y le damos "Create repository"

- ¿Cómo crear una rama en Git?

Para crear una nueva rama usamos el comando `git branch "nombre de la nueva rama"`

- ¿Cómo cambiar a una rama en Git?

Para moverse a otra rama usamos el comando `git checkout "nombre de la rama"`

- ¿Cómo fusionar ramas en Git?

Para fusionar rama usamos el comando `git merge "rama que fusionamos"` y se fusiona con la rama en la que estamos parados actualmente

- ¿Cómo crear un commit en Git?

Usamos el comando `git commit` podemos agregar al final `-m "mensaje"` para dejar un comentario

- ¿Cómo enviar un commit a GitHub?

Usamos `git push -u origin master` para poder enviar por primera vez el commit y después podemos usar solo `git push`

- ¿Qué es un repositorio remoto?

Es un repositorio guardado en internet

- ¿Cómo agregar un repositorio remoto a Git?

Lo podemos hacer con `git remote add origin "url del repositorio"`

- ¿Cómo empujar cambios a un repositorio remoto?

Usamos `git push -u origin main`

- ¿Cómo tirar de cambios de un repositorio remoto?

Usamos `git pull`

- ¿Qué es un fork de repositorio?

Es una copia de otro repositorio este fork no afecta al original cuando lo modificamos

- ¿Cómo crear un fork de un repositorio?

Estando en el repositorio en la parte derecha por encima de la descripción presionamos fork le damos un nombre y le damos a crear

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

En nuestro fork tenemos que darle a "code" y dentro tendremos la opción de pull request, después podemos elegir la rama en concreto y enviar el pull request

- ¿Cómo aceptar una solicitud de extracción?

En la pestaña de revisar cambios podremos aceptar pull mediante el botón que lo indica

- ¿Qué es una etiqueta en Git?

Es una rama que no se puede modificar

- ¿Cómo crear una etiqueta en Git?

Usamos `git tag "nombre"`

- ¿Cómo enviar una etiqueta a GitHub?

Se envían cuando usamos `git push`

- ¿Qué es un historial de Git?

Son todos los commit anteriores

- ¿Cómo ver el historial de Git?

Usamos `git reflog`

- ¿Cómo buscar en el historial de Git?

Usamos `git reset --"modo" hash-commit`

- ¿Cómo borrar el historial de Git?

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo puede ver y acceder solo el dueño del perfil

- ¿Cómo crear un repositorio privado en GitHub?

Cuando creamos un repositorio nos da la opción de hacerlo privado solo hay que seleccionarla

- ¿Cómo invitar a alguien a un repositorio privado en GitHub?

En el repositorio entramos en la configuración y en la parte de acceso usamos "Agregar personas" escribimos el nombre del usuario y aceptamos

- ¿Qué es un repositorio público en GitHub?

Es un repositorio cuyo acceso es libre para todos

- ¿Cómo crear un repositorio público en GitHub?

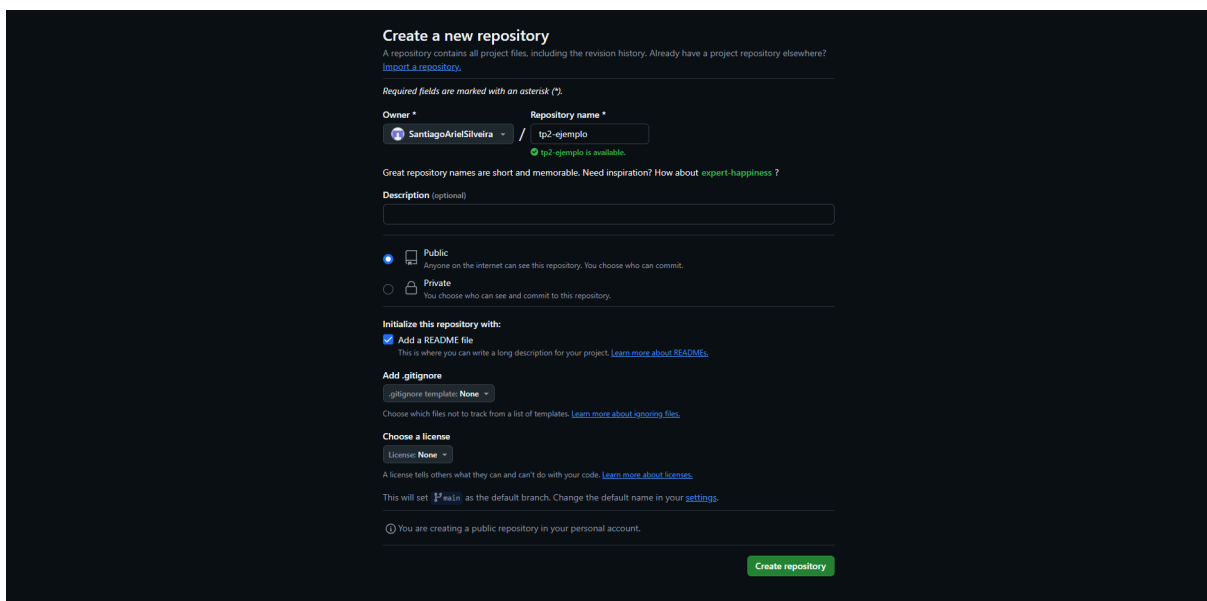
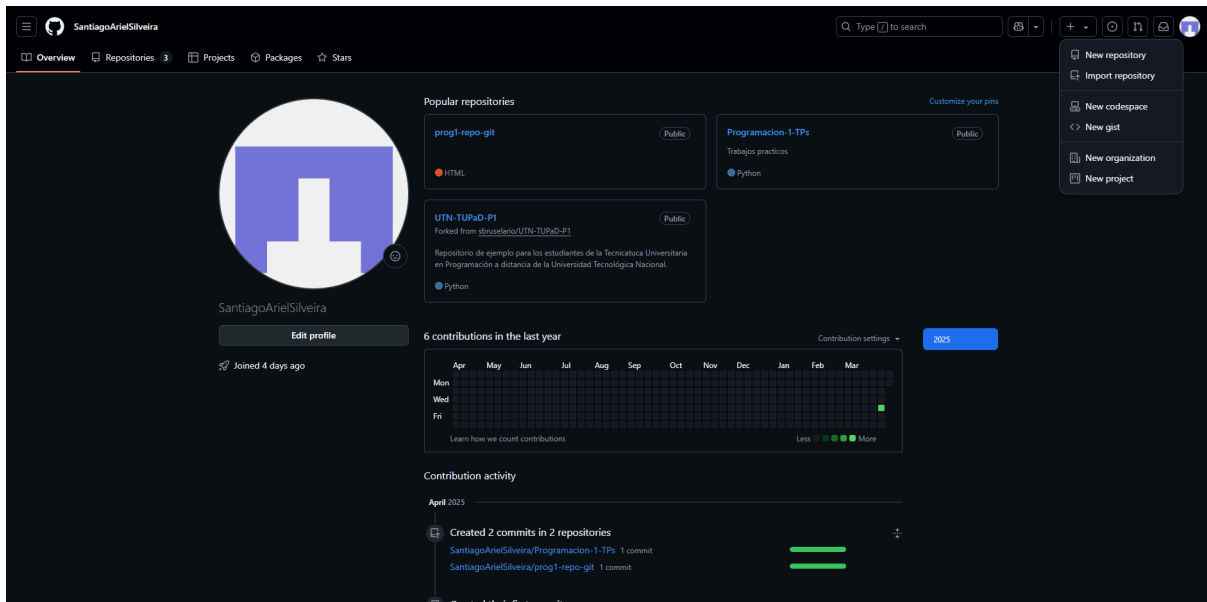
Cuando creamos un repositorio en GitHub la opción de "público" viene predefinidamente seleccionada, así que solo tenemos que darle un nombre y aceptar

- ¿Cómo compartir un repositorio público en GitHub?

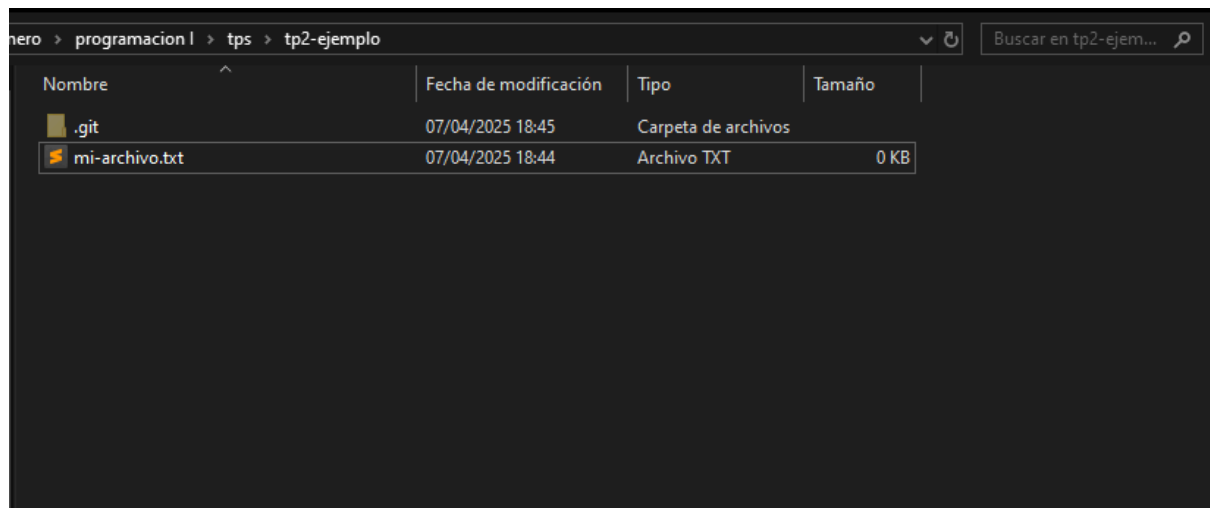
Podemos compartirlo a través de su URL o link con cualquiera

2) Realizar la siguiente actividad:

- Crear un repositorio.
 - o Dale un nombre al repositorio.
 - o Elije el repositorio sea público.
 - o Inicializa el repositorio con un archivo.



- Agregando un Archivo
 - o Crea un archivo simple, por ejemplo, "mi-archivo.txt".
 - o Realiza los comandos `git add .` y `git commit -m "Agregando mi-archivo.txt"` en la línea de comandos.
 - o Sube los cambios al repositorio en GitHub con `git push origin main` (o el nombre de la rama correspondiente).



```
MINGW64/f/escritorio/facu/primero/programacion I/tps/tp2-ejemplo
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps
$ git clone https://github.com/SantiagoArielSilveira/tp2-ejemplo
Cloning into 'tp2-ejemplo'...
warning: You appear to have cloned an empty repository.
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps
$ cd tp2-ejemplo
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/tp2-ejemplo (main)
$ git add .
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/tp2-ejemplo (main)
$ git commit -m "Agregando mi-archivo.txt"
[main (root-commit) 0238a91] Agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/tp2-ejemplo (main)
$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)
nothing to commit, working tree clean
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/tp2-ejemplo (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 231 bytes | 231.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SantiagoArielSilveira/tp2-ejemplo
 * [new branch] main -> main
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/tp2-ejemplo (main)
$
```

- Creando Branches
 - o Crear una Branch
 - o Realizar cambios o agregar un archivo
 - o Subir la Branch

```

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp
2-ejemplo (main)
$ git branch nueva-rama

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (main)
$ git branch
* main
  nueva-rama

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (main)
$ git checkout nueva-rama
Switched to branch 'nueva-rama'

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (nueva-rama)
$ git branch
  main
* nueva-rama

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (nueva-rama)
$ git add .

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (nueva-rama)
$ git commit -m "nueva rama"
[nueva-rama b9e9a0f] nueva rama
1 file changed, 1 insertion(+)

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (nueva-rama)
$ git status
On branch nueva-rama
nothing to commit, working tree clean

```

```

MINGW64:/f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (nueva-rama)
$ git push --all -u
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 270 bytes | 270.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nueva-rama' on GitHub by visiting:
remote:   https://github.com/SantiagoArielSilveira/tp2-ejemplo/pull/new/nueva-rama
remote:
To https://github.com/SantiagoArielSilveira/tp2-ejemplo
 * [new branch]      nueva-rama -> nueva-rama
branch 'main' set up to track 'origin/main'.
branch 'nueva-rama' set up to track 'origin/nueva-rama'.

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primeroprogramacion I/tps/tp2-ejemplo (nueva-rama)
$

```

SantiagoArielSilveira / tp2-ejemplo

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Branches

Overview Yours Active Stale All

Search branches...

Default

| Branch | Updated | Check status | Behind / Ahead | Pull request |
|--------|------------|--------------|----------------|--------------|
| main | 1 hour ago | | Default | |

Your branches

| Branch | Updated | Check status | Behind / Ahead | Pull request |
|------------|---------------|--------------|----------------|--------------|
| nueva-rama | 3 minutes ago | | 0 1 | |

Active branches

| Branch | Updated | Check status | Behind / Ahead | Pull request |
|------------|---------------|--------------|----------------|--------------|
| nueva-rama | 3 minutes ago | | 0 1 | |

3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

- Ve a GitHub e inicia sesión en tu cuenta.
- Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
- Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
- Opcionalmente, añade una descripción.
- Marca la opción "Initialize this repository with a README".
- Haz clic en "Create repository".

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

SantiagoArielSilveira / conflict-exercise

✔ conflict-exercise is available.

Great repository names are short and memorable. Need inspiration? How about **cuddly-fortnight** ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set **main** as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

Paso 2: Clonar el repositorio a tu máquina local

- Copia la URL del repositorio (usualmente algo como `https://github.com/tuusuario/conflict-exercise.git`).

- Abre la terminal o línea de comandos en tu máquina.
- Clona el repositorio usando el comando:
git clone https://github.com/tuusuario/conflict-exercise.git
- Entra en el directorio del repositorio:
cd conflict-exercise

```

MINGW64:/f/escritorio/facu/primero/programacion I/tps/conflict-exercise
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps
$ git clone https://github.com/SantiagoArielSilveira/conflict-exercise
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps
$ cd conflict-exercise

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflict-exercise (main)
$ |

```

Paso 3: Crear una nueva rama y editar un archivo

- Crea una nueva rama llamada feature-branch:
git checkout -b feature-branch
- Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:
Este es un cambio en la feature branch.
- Guarda los cambios y haz un commit:
git add README.md
git commit -m "Added a line in feature-branch"

```

README.md: Bloc de notas
Archivo Edición Formato Ver Ayuda
# conflict-exercise
nueva rama

```

```
MINGW64:/f/escritorio/facu/primero/programacion I/tps/conflict-exercise
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (feature-branch)
$ git add README.md

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch af8be75] Added a line in feature-branch
1 file changed, 2 insertions(+), 1 deletion(-)

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (feature-branch)
$ |
```

Paso 4: Volver a la rama principal y editar el mismo archivo

- Cambia de vuelta a la rama principal (main):

git checkout main

- Edita el archivo README.md de nuevo, añadiendo una línea diferente:

Este es un cambio en la main branch.

- Guarda los cambios y haz un commit:

git add README.md

git commit -m "Added a line in main branch"

README.md: Bloc de notas

Archivo Edición Formato Ver Ayuda

conflict-exercise

rama principal|

```
MINGW64:/f/escritorio/facu/primero/programacion I/tps/conflict-exercise
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$ git add README.md

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$ git commit -m "Added a line in main branch"
[main 2f40eba] Added a line in main branch
1 file changed, 2 insertions(+), 1 deletion(-)

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$
```


Paso 5: Hacer un merge y generar un conflicto

- Intenta hacer un merge de la feature-branch en la rama main:

git merge feature-branch

- Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

```
MINGW64:/f/escritorio/facu/primero/programacion I/tps/conflict-exercise
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$ git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main|MERGING)
$
```

Paso 6: Resolver el conflicto

- Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:

```
<<<<<<< HEAD
```

Este es un cambio en la main branch.

```
=====
```

Este es un cambio en la feature branch.

```
>>>>>>> feature-branch
```

- Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.
- Edita el archivo para resolver el conflicto y guarda los cambios (Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).
- Añade el archivo resuelto y completa el merge:

git add README.md

git commit -m "Resolved merge conflict"

```
README.md: Bloc de notas
Archivo Edición Formato Ver Ayuda
# conflict-exercise
<<<<<<< HEAD
rama principal
=====
nueva rama
>>>>>>> feature-branch
```

```
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$ git add README.md

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflct-exercise (main)
$ git commit -m "Resolved merge conflict"
[main a05c2bf] Resolved merge conflict
1 file changed, 1 insertion(+)
```

Paso 7: Subir los cambios a GitHub

- Sube los cambios de la rama main al repositorio remoto en GitHub:

git push origin main

- También sube la feature-branch si deseas:

git push origin feature-branch

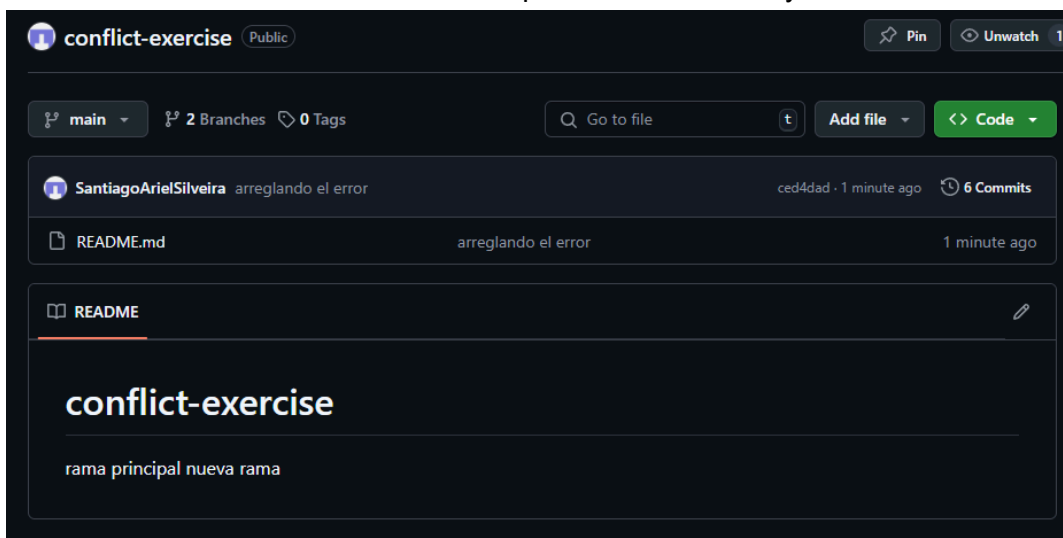
```
MINGW64:/f/escritorio/facu/primero/programacion I/tps/conflict-exercise
USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflict-exercise (main)
$ git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (12/12), 1.09 KiB | 371.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/SantiagoArielSilveira/conflict-exercise
   cea832b..a05c2bf  main -> main

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflict-exercise (main)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:   https://github.com/SantiagoArielSilveira/conflict-exercise/pull/new
remote:   /feature-branch
remote:
To https://github.com/SantiagoArielSilveira/conflict-exercise
 * [new branch]      feature-branch -> feature-branch

USUARIO@DESKTOP-CBVVTJM MINGW64 /f/escritorio/facu/primero/programacion I/tps/co
nflict-exercise (main)
$ |
```

Paso 8: Verificar en GitHub

- Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.
- Puedes revisar el historial de commits para ver el conflicto y su resolución.



(tenia un error en el readme asi que lo arregle con otro commit)